## AMENDMENTS TO THE CLAIMS

Kindly amend claims 1, 8, and 18 as shown in the following listing of claims. The listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims**

1. (Currently Amended) An apparatus in a microprocessor for executing native instructions that are provided directly to the microprocessor via an external instruction bus, the apparatus comprising:

instruction translation logic, configured to retrieve macro instructions provided via the external instruction bus, and configured to translate each of said macro instructions into associated native instructions for execution, wherein, if a first form of a first macro instruction is retrieved, said instruction translation logic directs the microprocessor to enable a native bypass mode and indicates such by asserting a first bit within a control register; and

bypass logic, coupled to said instruction translation logic, configured to access said first bit within said control register to determine if said native bypass mode has been enabled, and to detect wrapper macro instructions and, upon detection of said wrapper macro instructions, to disable said instruction translation logic, and to provide the native instructions for execution by the microprocessor, thereby bypassing said instruction translation logic;

wherein said wrapper macro instructions are existing macro instructions which are translated by said instruction translation logic according to architectural specifications if said native bypass mode has not been enabled.

2. (Original) The apparatus as recited in claim 1, wherein the native instructions are embedded within said wrapper instructions and are provided from a memory to the external instruction bus.

3. (Original) The apparatus as recited in claim 1, wherein the microprocessor employs a second bit within a flags register to indicate the occurrence of an interrupting event.

4. (Original) The apparatus as recited in claim 3, further comprising:

interrupt/exception/switch logic, configured to clear said first bit within said control register upon occurrence of said interrupting event and prior to transferring control to an interrupt event service routine, thereby disabling said native bypass mode, and configured to assert said second bit within said flags register, thereby indicating occurrence of said interrupting event.

5. (Original) The apparatus as recited in claim 4, wherein said flags register comprises an x86 EFLAGS register, and wherein said second bit comprises bit 31 within said x86 EFLAGS register.

6. (Original) The apparatus as recited in claim 1, wherein said bypass logic comprises:

a native instruction router, coupled to mode detection logic, configured to receive said wrapper macro instructions, and configured to strip the native instructions from within said wrapper macro instructions, and configured to route the native instructions to a native instruction bus.

7. (Original) The apparatus as recited in claim 1, wherein, if a second form of said first macro instruction is retrieved, said instruction translation logic directs the microprocessor to disable said native bypass mode and indicates such by clearing said first bit within said control register.

8. (Currently Amended) An apparatus, for allowing a micro instruction to be directly provided from an external instruction bus to execution logic within a pipeline microprocessor, the apparatus comprising:

a translator, for receiving macro instructions from a macro instruction bus, and for translating each of said macro instructions into associated micro instructions, said associated micro instructions being provided to the execution logic via a micro instruction bus; and

bypass logic, coupled to said translator, for routing the micro instruction to the execution logic, said bypass logic comprising:

a mode detector, for detecting a native bypass mode, and for detecting a wrapper macro instruction, and for directing that said translator cease instruction translation, wherein said wrapper macro instruction is an existing macro instructions which would otherwise be translated by said translation logic according to architectural specifications if said native bypass mode is not enabled.; and

native instruction routing logic, coupled to said mode detector, for receiving said wrapper macro instruction from said macro instruction bus, and for providing said micro instruction to said micro instruction bus, thereby circumventing said translator.

9. (Original) The apparatus as recited in claim 8, wherein the micro instruction is embedded within said wrapper macro instruction.

10. (Original) The apparatus as recited in claim 9, wherein said wrapper macro instruction comprises and x86 load effective address (LEA) instruction, and wherein said micro instruction is embedded within a 32-bit displacement field of said LEA instruction.

11. (Original) The apparatus as recited in claim 8, wherein said mode detector detects said native bypass mode by evaluating the state of a first bit within a control register.

12. (Original) The apparatus as recited in claim 11, wherein said first bit is asserted as a result of translating a first form of a first macro instruction.

13. (Original) The apparatus as recited in claim 12, wherein said first bit is cleared as a result of translating a second form of said first macro instruction.

14. (Original) The apparatus as recited in claim 13, wherein said first macro instruction comprises an invalid or spare macro instruction within an existing instruction set architecture to which the pipeline microprocessor conforms.

15. (Original) The apparatus as recited in claim 8, wherein the pipeline microprocessor employs a second bit within a flags register to indicate the occurrence of an interrupting event.

16. (Original) The apparatus as recited in claim 15, further comprising:

interrupt/exception/switch logic, configured to clear a first bit within a control register upon occurrence of said interrupting event and prior to transferring control to an interrupt event service routine, thereby disabling said native bypass mode, and configured to assert said second bit within said flags register, thereby indicating occurrence of said interrupting event.

17. (Original) The apparatus as recited in claim 15, wherein said flags register comprises an x86 EFLAGS register, and wherein said second bit comprises bit 31 within said x86 EFLAGS register.

18. (Currently Amended) A method for providing a plurality of native instructions stored in a memory directly to a microprocessor for execution, the method comprising:

enabling a native instruction bypass mode within the microprocessor;

embedding the plurality of native instructions within a corresponding plurality of wrapper instructions and providing the corresponding plurality of wrapper instructions to the microprocessor, wherein the corresponding plurality of wrapper instructions are existing macro instructions, and wherein the corresponding plurality of wrapper instructions would otherwise be translated by instruction translation logic according to architectural specifications in the absence of said enabling; and

within the microprocessor, detecting the native instruction bypass mode and extracting the plurality of native instructions from within the corresponding plurality of wrapper instructions.

19. (Original) The method as recited in claim 18, wherein said enabling comprises:

detecting a first form of an otherwise invalid or spare macro instruction; and

asserting a bit within a control register to indicate that the microprocessor is in a the native instruction bypass mode.

20. (Original) The method as recited in claim 18, wherein said embedding comprises:

encoding each of the plurality of native instructions within a field of the corresponding plurality of wrapper instructions, wherein each of the corresponding plurality of wrapper instructions comprises an otherwise valid macro instruction.

21. (Original) The method as recited in claim 20, wherein the otherwise valid macro instruction comprises an x86 load effective address instruction, and wherein the field comprises a displacement field.

22. (Original) The method as recited in claim 18, further comprising:

upon detection of an interrupting event, asserting a bit within a flags register and disabling the native instruction bypass mode; and

upon return from an interrupting event, evaluating the bit within the flags register and re-enabling the native instruction bypass mode.

23. (Original) The method as recited in claim 19, wherein the flags register comprises an x86 EFLAGS register, and where the bit comprises bit 31 within the x86 EFLAGS register.